UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/651,376	08/29/2003	Shubhendu S. Mukherjee	42P15452	9255

8791 7590 10/24/2006

BLAKELY SOKOLOFF TAYLOR & ZAFMAN
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CA 90025-1030

EXAMINER	
EHNE, CHARLES	

ART UNIT	PAPER NUMBER
2113	

DATE MAILED: 10/24/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/651,376	MUKHERJEE ET AL.	
	Examiner	Art Unit	
	Charles Ehne	2113	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *08 August 2006*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-29* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-29* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 1,3,10,11,14,18-20,22,23 and 27-29 are rejected under 35 U.S.C. 102(b) as being unpatentable by Bossen (6,058,491).

As to claim 1, Bossen discloses a method comprising:

executing corresponding instruction threads as a leading thread and a trailing thread (column 3, lines 1-5);

saving a processor state corresponding to execution of a selected instruction in a history buffer before writing a result from the selected instruction to a destination register (column 4, lines 61-63);

comparing the result from the selected instruction executed in the leading thread to the result from the selected instruction executed in the trailing thread (column 4, lines 61-66); and

restoring the processor state corresponding to a previous instruction using data from the history buffer if the comparison indicates a fault (column 4, lines 61-66).

As to claim 3, Bossen discloses the method of claim 1 wherein the leading thread and the trailing thread are executed by multiple processors (column 2, lines 61-63).

As to claim 10, Bossen discloses the method of claim 1 wherein the selected instruction comprises a branch instruction (column 5, lines 46-48).

As to claim 11, Bossen discloses an apparatus comprising:

means for executing corresponding threads as a leading thread and a trailing thread (column 3, lines 1-5);

means for saving a processor state corresponding to execution of a selected instruction before writing a result from the selected instruction to a destination register (column 4, lines 61-63);

means for comparing the result from the selected instruction executed in the leading thread to the result from the selected instruction executed in the trailing thread (column 4, lines 61-66); and

means for restoring the processor state corresponding to a previous instruction if the comparison indicates a fault (column 4, lines 61-66).

As to claim 14, Bossen discloses an apparatus comprising:

leading thread execution circuitry to execute a leading thread of instructions (column 3, lines 1-5);

trailing thread execution circuitry to execute a trailing thread of instructions (column 3, lines 1-5); and

a history buffer coupled with the leading thread execution circuitry and the trailing thread execution circuitry to store information related to execution of a selected instruction from the leading thread of instructions, wherein the information stored in the

history buffer is used to restore an architectural state corresponding to a checkpoint if an execution fault is detected (column 4, lines 61-66).

As to claim 18, Bossen discloses the apparatus of claim 14 wherein the execution fault is caused by a branch instruction (column 6, lines 30-34).

As to claim 19, Bossen discloses the apparatus of claim 14 wherein the checkpoint corresponds to the architectural state at a time at which an instruction causing the fault is started (column 3, lines 19-29).

As to claim 20, Bossen discloses the apparatus of claim 14 wherein the checkpoint corresponds to the architectural state at a time prior to which an instruction causing the fault is started (column 7, lines 21-26).

As to claim 22, Bossen discloses the apparatus of claim 14 wherein the leading thread execution circuitry is part of a first processor and the trailing thread execution circuitry are part of a second processor (column 2, lines 61-63).

As to claim 23, Bossen discloses a system comprising:

leading thread execution circuitry to execute a leading thread of instructions (column 3, lines 1-5);

trailing thread execution circuitry to execute a trailing thread of instructions (column 3, lines 1-5);

an input/output controller coupled with the leading thread execution circuitry (column 5, lines 1-3); and

a history buffer coupled with the leading thread execution circuitry and the trailing thread execution circuitry to store information related execution of a selected instruction from the leading thread of instructions, wherein the information stored in the history buffer is used to restore an architectural state corresponding to a checkpoint if an execution fault is detected (column 4, lines 61-66).

As to claim 27, Bossen discloses the system of claim 23 wherein the execution fault is caused by a branch instruction (column 6, lines 30-34).

As to claim 28, Bossen discloses the system of claim 23 wherein the checkpoint corresponds to the architectural state at a time at which an instruction causing the fault is started (column 3, lines 19-29).

As to claim 29, Bossen discloses the system of claim 23 wherein the leading thread execution circuitry is part of a first processor and the trailing thread execution circuitry are part of a second processor (column 2, lines 61-63).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

Claims 2 and 21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bossen taken in view of Mukherjee (2001/0034854).

As to claims 2 and 21 Bossen discloses a method comprising: executing corresponding instruction threads as a leading thread and a trailing thread (column 3, lines 1-5); saving a processor state corresponding to execution of a selected instruction in a history buffer before writing a result from the selected instruction to a destination register (column 4, lines 61-63); comparing the result from the selected instruction executed in the leading thread to the result from the selected instruction executed in the trailing thread (column 4, lines 61-66); and restoring the processor state corresponding to a previous instruction using data from the history buffer if the comparison indicates a fault (column 4, lines 61-66). Bossen fails to disclose wherein the leading thread and the trailing thread are executed by a single processor.

Mukherjee discloses a simultaneous and redundantly threaded, pipelined processor that executes the same set of instructions simultaneously as two separate threads to provide fault tolerance (Abstract: lines 1-3). Mukherjee does disclose

wherein the leading thread and the trailing thread are executed by a single processor (page 4, ¶ 0039, lines 5-11).

It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to implement Mukherjee's single processor with Bossen's method for comparing the results of leading and trailing threads. A person of ordinary skill in the art would have been motivated to make the modification because Mukherjee's SRT processor detects transient faults without sever penalties and additional component (page 4, ¶ 0039, lines 1-4) .

Claims 4-9,12,13,15-17 and 24-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bossen taken in view of Ando (6,519,730).

As to claims 4,15 and 24 Bossen discloses a method comprising: executing corresponding instruction threads as a leading thread and a trailing thread (column 3, lines 1-5); saving a processor state corresponding to execution of a selected instruction in a history buffer before writing a result from the selected instruction to a destination register (column 4, lines 61-63); comparing the result from the selected instruction executed in the leading thread to the result from the selected instruction executed in the trailing thread (column 4, lines 61-66); and restoring the processor state corresponding to a previous instruction using data from the history buffer if the comparison indicates a fault (column 4, lines 61-66). Bossen fails to disclose wherein the processor state is stored in an entry in the history buffer that stores an instruction pointer to the selected instruction, a value stored in the destination register, wherein the value in the destination register is to be overwritten by the result of the selected instruction, and a

register map that indicates a mapping of one or more architectural registers to one or more physical registers.

Ando discloses a method of restoring the state of a processor to an earlier state before the fault was detected (column 3, lines 53-57). Ando does disclose wherein the processor state is stored in an entry in the history buffer that stores an instruction pointer to the selected instruction, a value stored in the destination register, wherein the value in the destination register is to be overwritten by the result of the selected instruction, and a register map that indicates a mapping of one or more architectural registers to one or more physical registers (columns 3-4, lines 64-17).

It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to implement Ando's history buffer that stores an instruction pointer to the selected instruction, a value stored in the destination register, wherein the value in the destination register is to be overwritten by the result of the selected instruction, and a register map that indicates a mapping of one or more architectural registers to one or more physical registers with Bossen's history buffer. A person of ordinary skill in the art would have been motivated to make the modification because the checkpoint array provides pointers which allow instructions to be retried by returning to the previous correct state. (Ando: column 4, lines 5-12).

As to claim 16, Ando discloses the apparatus of claim 15 wherein the architectural state corresponding to the checkpoint is restored by selectively flushing results of instructions that started execution after an instruction causing the fault started execution and restoring architectural state to a checkpoint corresponding to a state at

which the instruction causing the fault started execution, wherein at least a portion of the restored architectural state is retrieved from the history buffer (column 6, lines 37-39 & column 4, lines 51-59).

As to claim 17, Ando discloses the apparatus of claim 16 wherein the architectural state corresponding to the checkpoint is restored by flushing non-retired speculative instructions from the execution circuitry corresponding to the thread having an instruction that caused the fault, flushing an architectural state of the execution circuitry corresponding to the thread having the instruction that caused the fault, and flushing the history buffer after register values used to restore the architectural state to the checkpoint are retrieved (column 6, lines 37-39 & column 4, lines 51-59).

As to claim 25, Ando discloses the system of claim 24 wherein the architectural state corresponding to the checkpoint is restored by selectively flushing results of instructions that started execution after an instruction causing the fault started execution and restoring architectural state to a checkpoint corresponding to a state at which the instruction causing the fault started execution, wherein at least a portion of the restored architectural state is retrieved from the history buffer (column 6, lines 37-39 & column 4, lines 51-59).

As to claim 26, Ando discloses the system of claim 25 wherein the architectural state corresponding to the checkpoint is restored by flushing non-retired speculative instructions from the execution circuitry corresponding to the thread having an instruction that caused the fault, flushing an architectural state of the execution circuitry corresponding to the thread having the instruction that caused the fault, and flushing the

history buffer after register values used to restore the architectural state to the checkpoint are retrieved (column 6, lines 37-39 & column 4, lines 51-59).

As to claims 5 and 12 Bossen discloses a method comprising: executing corresponding instruction threads as a leading thread and a trailing thread (column 3, lines 1-5); saving a processor state corresponding to execution of a selected instruction in a history buffer before writing a result from the selected instruction to a destination register (column 4, lines 61-63); comparing the result from the selected instruction executed in the leading thread to the result from the selected instruction executed in the trailing thread (column 4, lines 61-66); and restoring the processor state corresponding to a previous instruction using data from the history buffer if the comparison indicates a fault (column 4, lines 61-66). Bossen fails to disclose wherein restoring the processor state corresponding to a previous instruction if the comparison indicates a fault comprises: selectively flushing results of instructions that started execution after an instruction causing the fault started execution; restoring architectural state to a checkpoint corresponding to a state at which the instruction causing the fault started execution, wherein at least a portion of the restored architectural state is retrieved from the history buffer.

Ando discloses a method of restoring the state of a processor to an earlier state before the fault was detected (column 3, lines 53-57). Ando does disclose wherein restoring the processor state corresponding to a previous instruction if the comparison indicates a fault comprises: selectively flushing results of instructions that started execution after an instruction causing the fault started execution (column 4, lines 51-59);

restoring architectural state to a checkpoint corresponding to a state at which the instruction causing the fault started execution, wherein at least a portion of the restored architectural state is retrieved from the history buffer (columns 3-4, lines 53-12).

It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to implement Ando's selectively flushing results of instructions and restoring architectural state to a checkpoint corresponding to a state at which the instruction causing the fault started execution, wherein at least a portion of the restored architectural state is retrieved from the history buffer with Bossen's history buffer. A person of ordinary skill in the art would have been motivated to make the modification because Ando's flushing results that occurred after the fault a allows for the processors to retry the previous instructions (Ando: column 4, lines 13-17).

As to claim 6, Ando discloses the method of claim 5 wherein selectively flushing results of instructions that started execution after an instruction causing the fault started execution comprises:

flushing non-retired speculative instructions from the leading thread execution circuitry (column 6, lines 37-39 & column 4, lines 51-59);

flushing an architectural state of the trailing thread from the trailing thread execution circuitry (column 6, lines 37-39 & column 4, lines 51-59); and

flushing the history buffer after register values used to restore the architectural state to the checkpoint are retrieved (column 4, lines 8-17).

As to claim 7, Ando discloses the method of claim 5 wherein the checkpoint corresponds to the architectural state at a time at which execution of the instruction causing the fault is started (column 4, lines 8-12).

As to claim 8, Ando discloses the method of claim 5 wherein the checkpoint corresponds to the architectural state at a time prior to which execution of the instruction causing the fault is started (column 4, lines 8-12).

As to claim 9, Ando discloses the method of claim 5 wherein restoring architectural state to a checkpoint corresponding to a state at which the instruction causing the fault started execution, wherein at least a portion of the restored architectural state is retrieved from a history buffer comprises:

retrieving previous register values for registers written by or subsequent to the instruction causing the fault from the history buffer for the leading thread (column 4, lines 62-67);

restoring the previous register values to provide a restored architectural state for the leading thread (column 4, lines 62-67); and

copying the restored architectural state for the leading thread to an architectural register file for the trailing thread to provide a restored architectural state for the trailing thread (column 6, lines 37-44 & columns 3-4 and lines 64-4)).

As to claim 13 Bossen discloses a method comprising: executing corresponding instruction threads as a leading thread and a trailing thread (column 3, lines 1-5); saving a processor state corresponding to execution of a selected instruction in a history buffer before writing a result from the selected instruction to a destination register (column 4,

lines 61-63); comparing the result from the selected instruction executed in the leading thread to the result from the selected instruction executed in the trailing thread (column 4, lines 61-66); and restoring the processor state corresponding to a previous instruction using data from the history buffer if the comparison indicates a fault (column 4, lines 61-66). Bossen fails to disclose wherein the means for selectively flushing results of instructions that started execution after an instruction causing the fault started execution comprises: means for flushing non-retired speculative instructions from a thread having the instruction that caused the fault; means for flushing an architectural state of a trailing thread having the instruction that caused the fault; and means for flushing a history buffer after register values used to restore the architectural state to the checkpoint are retrieved.

Ando discloses a method of restoring the state of a processor to an earlier state before the fault was detected (column 3, lines 53-57). Ando does disclose wherein the means for selectively flushing results of instructions that started execution after an instruction causing the fault started execution comprises: means for flushing non-retired speculative instructions from a thread having the instruction that caused the fault (column 6, lines 37-39 & column 4, lines 51-59); means for flushing an architectural state of a trailing thread having the instruction that caused the fault (column 6, lines 37-39 & column 4, lines 51-59); and means for flushing a history buffer after register values used to restore the architectural state to the checkpoint are retrieved (column 4, lines 8-17).

It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to implement Ando's selectively flushing results of instructions and restoring architectural state to a checkpoint corresponding to a state at which the instruction causing the fault started execution, wherein at least a portion of the restored architectural state is retrieved from the history buffer with Bossen's history buffer. A person of ordinary skill in the art would have been motivated to make the modification because Ando's flushing results that occurred after the fault a allows for the processors to retry the previous instructions (Ando: column 4, lines 13-17).

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Charles Ehne whose telephone number is (571)-272-2471. The examiner can normally be reached on Monday-Friday 8:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Robert Beausoliel can be reached on (571)-272-3645. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.